API Design:

The api design, as I understand it, you guys are using Lambda functions pointing at backend redshift database. A single lambda could be written with simple logic circuit based on the parameters included. I’m describing two below instead to separate and keep smaller. The concern here would be that some of these aren’t tactical queries, using lambda and its 15 minute time limit is a consideration. The tables we’re generating are well-summarised and should work in this context but using API for decision source reporting is somewhat dodgy. At least I’ve seen plenty of queries that didn’t return in 15 minutes.

While I haven’t written of these professionally, the gist of it is we take in HTTP request, parse the parameters, find the sql associated, substitute parameters, submit to DB, return results in a return HTTP object. That is the way of the restAPI and we are doin it with data.

IN theory we could have a single script run all of it. Either way we’ll start with a top level identifier:

**Type** with values (‘ProductionCompany’,’Genre’)

**Production Company**

Three parameters

Production Company (“ALL”, 1 value, list of multiple) optional

**Year** with values (‘ALL’,[Year -n]) customize to allow for ranges with (2001:2004) notation

**Metric** with values (‘Budget’,’Revenue’,’Profit’,’GenreCount’,’Popularity’,’ALL’)

Combinations of all can be considered separated with ‘|’ as in “Budget|Revenue”

**Genre**

Four parameters

**Genre** with values (“ALL”, 1 value, list of multiple) optional

**Year** with values (‘ALL’,[Year -n]) allow for ranges with (2001:2004) notation

**Metric** with values (‘Budget’,’Revenue’,’Profit’,’GenreCount’,’Popularity’,’ALL’)

Combinations of all can be considered separated with ‘|’ as in “Budget|Revenue”

**PopularityValues** (optional) Will filter for the popularity ranking < N . No value defaults to 1

Other methods in addition to the Lamda functionality include using Flask or other Python web apps to create restful API’s and host them locally or remotely. If we knew we’re asking crazy questions we’d have to do something like this.

Scaling would definitely be more flexible in a cloud based solution. Monitoring in the cloud is essentially available for you. All files populating would be archived in an appropriate manner. Using S3 storage types we can quickly move the old files into cold storage for cost efficiency. Failure recovery on Amazon is what they do. Replication across regions if necessary is available. Finally the restAPI framework uses keys but can be made keyless if the network is internal. I’ve seen apis with no authentication required.